



Universidad  
Carlos III de Madrid



This is a postprint version of the following published document:

Larrabeiti, D., Kazovsky, L., Rodríguez, G., Aparicio, R., Shunrong Shen, T. & Yin, S. (2015). Integrating a next-generation optical access network testbed into a large-scale virtual research testbed. *2015 17th International Conference on Transparent Optical Networks (ICTON)*, Budapest (Hungria), pp. 1-6.

**DOI:** [10.1109/ICTON.2015.7193424](https://doi.org/10.1109/ICTON.2015.7193424)

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works

# Integrating a Next-Generation Optical Access Network Testbed into a Large-Scale Virtual Research Testbed

David Larrabeiti, Leonid Kazovsky\*, Gerson Rodríguez, Raquel Aparicio, Thomas Shunrong Shen\*, and Shuang Yin\*

*Universidad Carlos III de Madrid, Avda. Universidad 30, E-28911 Leganés, Spain*

*\*Stanford University, Stanford, USA*

*Tel. (+34) 91 624 9953, Fax: (+34) 91 624 8749, e-mail: dlarra@it.uc3m.es*

## ABSTRACT

Several experimental research networks have been created in the laboratories of prominent universities and research centres to assess new optical communication technologies. A greater value and research impact can be obtained from these testbeds by making them available to other researchers through research infrastructure federations such as GENI and/or FIRE. This is a challenging task due to the limitations of programmability of resource management and virtualisation software in most experimental optical networks. Fed4FIRE is an

EU research project that makes it possible to create complex testbed scenarios that interconnect heterogeneous testbeds distributed physically all over the world. In this paper, we present a practical approach for the federation of a next-generation optical access testbed created at Stanford University called UltraFlow

Access. That testbed offers its users both packet-switched and circuit-switched services while remaining compatible with conventional PONs. Our approach facilitates experimentation on the UltraFlow Access testbed in the context of large virtual testbeds using Fed4FIRE protocols.

## Keywords

FIRE, Fed4FIRE, UltraFlow access, optical access network, passive optical network.

## 1. INTRODUCTION

Novel networking concepts are being created in research laboratories all over the world, and demonstrated in testbeds whose utilization is constrained to local researchers. Usually, these network testbeds involve unique prototype components and measurement equipment. A wider impact and analysis of the possibilities of the new technology could be achieved if these testbeds are open to other researchers through an advanced Internet service. Moreover, the federation of a large amount of disperse testbed facilities can make it possible for multiple researchers to experiment with future Internet protocols and applications on a big computation grid for a given period of time. This is one of the goals of GENI (Global Environment for Network Innovations) in USA and FIRE (Future Internet Research and Experimentation) in Europe. Both initiatives aim to create a large-scale virtual laboratory for networking and distributed systems research and education.

In this paper we attempt to shed some light on the key aspects and procedures required to integrate an optical networking research testbed into a multi-technology environment by adopting software and protocols created in those projects that have become a de facto standard for open experimentation.

The paper is organised as follows. In Section 2, the reader is introduced to the Fed4Fire federation concept, which is the approach used in our work. The testbed to be federated – Stanford's UltraFlow Access Network – is described in Section 3. Then, Section 4 discusses key aspects of integration of the testbed within Fed4Fire and the software tools selected to manage it and to run network experiments. Section 5 gives a simplified view of what an external user would see through the portal. Finally, Section 6 presents our conclusions.

## 2. FEDERATION OF HETEROGENOUS RESEARCH NETWORK TESTBEDS: FED4FIRE

A number of projects aim to build a cross-national facility to enable experimentally-driven research in different parts of the world. Most of them are focused on a single research community. Fed4FIRE (2012-2016) is an Integrating Project under the European Union's Seventh Framework Programme (FP7) addressing the work programme topic Future Internet Research and Experimentation. The Fed4FIRE project intends to build an open, accessible and reliable framework for the federation of Internet research infrastructure across community borders. Sample community domains include optical networking, wireless networking, software defined networking, cloud computing, grid computing, smart cities, etc. All these heterogeneous communities, where we aim to include a next-generation optical access system, are involved in the project to guarantee compatibility and support of heterogeneous infrastructure and experimentation requirements.

The keys of this target framework are: 1) open experiment lifecycle management software; 2) experimental measurement and monitoring tools; 3) trust and security mechanisms; and 4) advanced inter-testbed connectivity services. Therefore, the federation of optical networking testbeds is not an easy goal. Most optical laboratory devices do not feature the programmability required to embed federation software, and lack the virtualisation capabilities required to enable secure access to the infrastructure to external users. Consequently, those elements of a testbed that do not support this functionality need to be assisted by a computer. We focus on this scenario

and try to integrate an optical access network testbed as powerful as possible. We attempt to integrate a next-generation optical access network prototype developed at Stanford University. The integration is possible via a collaboration effort between Fed4Fire and the PNRL group led by professor Leonid Kazovsky.

### 3. THE FEDERATED TESTBED: ULTRAFLOW ACCESS TESTBED

Optical Flow Switching (OFS) has been proposed as an alternative to hop-by-hop electronic packet switching for large data transfers, as it can feature very low end-to-end latency in the data transfer phase at ultra-broadband speeds. In OFS dedicated *lightpaths* are scheduled and dynamically allocated along the entire network path between end-systems [1]. No buffering occurs between the source and destination as packet switching is replaced by dynamic circuit switching. OFS *lightpaths* or equivalent electronic paths are scheduled (for an envisioned time period larger than 100 ms) using centralized network edge schedulers coordinated through an electronic control plane [1-2]. Efficient OFS network design and wavelength allocation mechanisms in the Metro/Core network have been investigated [2]. However, the design of an OFS-enabled access network, which bridges the end-user premises with the OFS Metro/Core, nor the possibilities of OFS as enabler of new services in the residential access area, have been thoroughly studied. Indeed, future networks are envisioned to bring a proliferation of networked services that will require ultra-low latency [3]. There is a need to find a way to deliver such ultra-low latency services by pushing cloud services toward the user in the access-metro area. UltraFlow Access [4] aims to achieve that goal. UltraFlow Access is a novel optical access network architecture that enables the coexistence of OFS and regular IP services over the same optical access network.

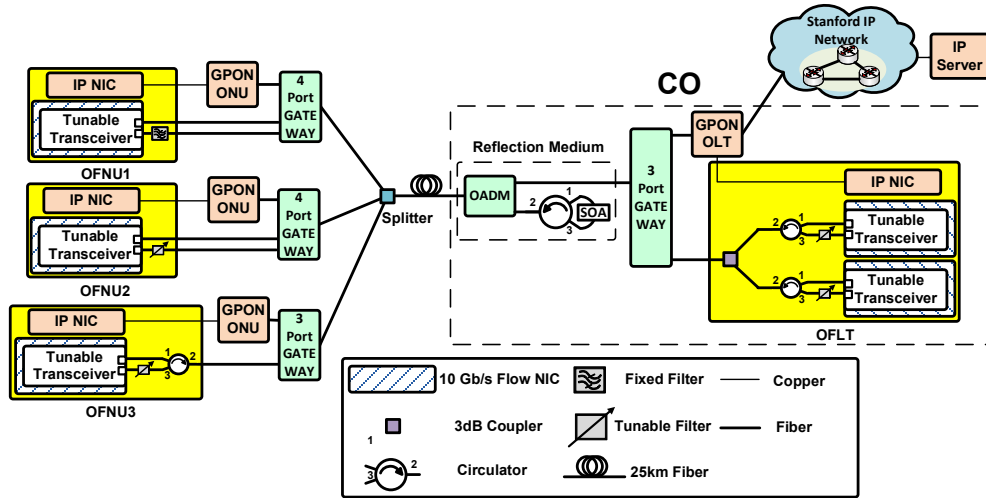


Figure 1. The UltraFlow Access testbed at Stanford University [3].

As illustrated in Fig. 1, Stanford's UltraFlow Access enables IP access using legacy Passive Optical Networks (e.g., GPON, G-EPON), and OFS access via a set of novel Optical Flow Network Units (OFNUs), which serve as Flow users/ servers, and are connected via the same infrastructure as PON, to a novel Optical Flow Line Terminal (OFLT) located at the central office (CO). The OFLT serves as interface between the Flow access and metro/core networks and it performs dynamic wavelength allocation (DWA) in the Access, and provides network-monitoring functions such as power and wavelength characterization. It also connects to the legacy IP path via the PON's optical line terminal (OLT). We designed and experimented three different OFNU architectures; each of them is equipped with a 10 Gbps tunable transceiver, a Flow network interface card (NIC) and an IP NIC connecting to a PON via an Optical Network Unit (ONU). The main difference between the three OFNUs is in the optical filter design. OFNU1 has a fixed optical filter, and thus it is a colored architecture (i.e., can receive on one specific wavelength only); this is cost-effective, but lacks flexibility in terms of scheduling and resource availability. A colorless OFNU design (i.e., can receive/transmit on any wavelength) is more costly, but it enables efficient resource utilization and scheduling flexibility. Thus, we consider two colorless OFNU architectures: OFNU2 and OFNU3. The architecture OFNU2 has a tunable filter covering the C-band to offer wavelength selection flexibility for downstream transmission. Similarly, OFNU3 employs a circulator to separate upstream and downstream Flow traffic, which also offers full wavelength flexibility for bidirectional transmission. To separate/combine IP and downstream/upstream Flow traffic based on different wavelengths utilized in both services, we have designed and implemented 3-port and 4-port gateways (see Fig. 1).

As shown in Fig. 1, two different types of communications are supported in the UltraFlow Access testbed: OFLT-OFNU and OFNU-OFNU. The OFLT-OFNU communication is the regular network to users communication (i.e., between the service provider and customers' premises); whereas the OFNU-OFNU communication is between two OFNUs located in the same Access Network. The latter is enabled through a Reflection medium located in the CO, where a specific upstream signal is reflected back into the downstream

direction via a circulator. A Semiconductor Optical Amplifier (SOA) is used to compensate for the “double-distance” fiber loss. Multicast can natively be enabled for both types of communications due to the star nature of the passive splitter.

#### 4. INTEGRATION OF ULTRAFLOW ACCESS IN FED4FIRE

Fed4Fire proposes the use of certain tools for basic federation implementing the Slice-based Facility Architecture (SFA), like the GENI Aggregate Manager API, experiment control [7][8], measurement collection [9] and monitoring [10] of both infrastructure/resources and experiments. These software components must be included in UltraFlow Access nodes permitting remote access and Fed4Fire includes a comprehensive guide and methodology for testbed integration.

##### 4.1 Architectural Extensions for Testbed Integration

The changes introduced to the UltraFlow Access testbed are shown in Fig. 2. To allow both Stanford University and external experimenters to carry out their experiments without interference, two Virtual Machines (VMs) have been enabled at two different nodes (pnrl05 and pnrl06) of the original testbed. These two virtual machines are managed by Emulab [11]. A new computer has been added to host two virtual machines that contain the Emulab servers (Emulab boss and Emulab ops). In addition to this server, a dedicated switch has also been added to give the VMs access to the Internet and to the Emulab servers. This configuration allows to isolate the Emulab-managed part from the rest of the network. This is done because Emulab has its own network management, which makes it more convenient to isolate this part of our network from the rest. The VMs have access to the optical transceivers of the testbed.

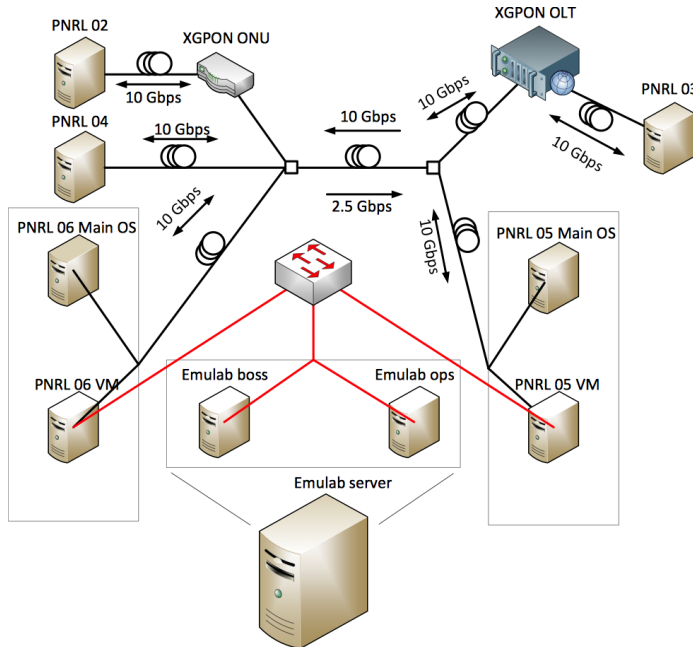


Figure 2. Architectural changes on the UltraFlow Access testbed for the Fed4Fire Federation.

##### 4.2 Basic Node and Testbed Management with Emulab

One of the first steps to address is to install an environment capable of managing the nodes in an automated fashion and basic experiment support. UltraFlow Access resources are virtualized in the sense that they may be allocated and controlled regardless of where the resource or the experimenter is physically located. To achieve this, UltraFlow testbed relies on Emulab [11], developed at Utah University. As this software is designed for environments with several nodes, it requires the use of at least one switch; thus, testbed administrators have to make a thorough design of the testbed network prior to installation. Testbeds using Emulab software usually offer the possibility of arbitrarily interconnecting their resources through the switch via Virtual LANs as requested by the experimenter. However, in the case of UltraFlow testbed, all reservable resources are already interconnected through optical equipment. Moreover, what they actually do is to act as entry points that give access to and allow to make use of this optical equipment, which is the real heart of the testbed. In addition to this internal interface, nodes also include a public one so that they can be configured by Emulab software, accessed by experimenters or connected to any resource at another testbed. Once Emulab is installed, a web interface for UltraFlow’s own Emulab becomes accessible so that the following tasks become possible: (a) administrators can configure resource types and check the testbed status; (b) new users can request a local account to the administrator; and c) users can create/delete/swap in/swap out experiments.

For the experimenters' convenience, a customised Ubuntu 12.04 image is included with the testbed. This image includes basic tools to generate (tcpdump, iperf, mgen, vlc, etc.) network traffic. In addition, since it is impossible to access the raw hardware from the VMs, a set of the scripts to read and write parameters from/to the optical transceivers are provided. This set of scripts allows to read the tx/rx power, the transmission channel of the network card, etc. The scripts connect to the hypervisor machines to read and write the desired parameters. They are designed in such a way that from any of the VMs it is possible for the experimenter to read and write parameters from both hypervisor machines indistinctly. For instance, it is possible to issue a command to the card in the *pnrl06* VM from the *pnrl05* VM. Finally, OMF and OML are also included in the default OS image. OMF and OML are covered in subsequent sections.

### 4.3 Experiment Management and Control

Once the basic node functionality is provided, it is also necessary to provide a software framework to allow the user to perform automated experiments in a user-friendly way. In this section, two possible ways to perform such experiments are presented. One of them is to use the Emulab interface directly and another one is to provide it through OMF, being the latter the recommended option by Fed4Fire.

#### a. Emulab experiment management

When a resource type has been configured and a device of this type is connected to the testbed, Emulab recognizes it and takes it over. From that moment on, administrators and authorized users (the user in control of the experiment the resource is in or on other authorized user in the project) can perform tasks on it through the web or by means of a terminal to the Emulab system, using Emulab-specific commands. For example, users can reserve/free a node when swapping an experiment in/out, reboot it, load a certain operating system or create images of its current status for future use or use on other nodes, if under their control. When a resource is occupied by an experiment, experimenters can also access the node via a console and use it with administrator privileges.

The experiment life cycle starts with an Emulab user requesting a new project via the web interface by filling a form. This project has to be approved by the testbed administrator and then, after it is created, the user becomes a temporary project leader, to whom privileges to authorise other users into the project are delegated. When it comes to reservations, Emulab offers experimenters (users that are creating an experiment within a project) either the use of ns syntax-like files or a graphical tool. Node types, operating system to load, IP address configuration and some other characteristics can be specifically requested. On the Stanford UltraFlow testbed, there is one unique node type *pc\_ultraflow*. Specifically, virtual machine *pc1* runs on host *pnrl05*, which uses an auto tunable receiver that gets automatically tuned to the manually tunable transmitter at host *pnrl06*, running virtual machine *pc2*.

#### b. OMF Experiment control

OMF is a framework provides a set of tools to describe, instrument and execute experiments and collect their results and a set of services to manage and operate testbed resources. This framework basically requires an experiment controller (EC) that processes an OEDL description of the experiment scenario and controls the required nodes and a resource controller (RC) daemon at every resource, which receives messages from the EC and executes the commands. Messaging between the EC and the RCs is performed via an XMPP server or an AMQP server.

### 4.4 Measurement Collection: the Orbit Management Library (OML)

Another important aspect that is tightly related to experiment and resource control is measurement collection. Our testbed provides this feature through OML. OML is a generic software framework for measurement collection [13]. It allows the experimenter to define the so called measurement points (MPs) inside new or pre-existing applications so that these MPs generate measurement streams (MSs) that can be directed to a central collection point, and stored in measurement databases. It is composed of two entities. The first one is an OML server responsible for collecting and storing measurements inside an SQLite3 or PostgreSQL database. The UltraFlow Access testbed offers a local SQLite3 database so that experimenters can direct their measures there. The second one is an OML client library that essentially provides a C API to be used inside applications. There are also native implementations in Python (OML4Py) and Ruby (OML4R) as well as third party libraries for Java (OML4J) and Javascript/WebSocket (OML4JS). The OML group offers both OML-enabled applications and good tutorials for OML client application programming.

### 4.5 Testbed Monitoring: Collectd

Collectd is a daemon running at every UltraFlow Access resource, which collects system performance statistics in a periodic manner by means of enabling different plugins: CPU, disk, interfaces, ip-tables, IRQ, memory, processes, protocols, uptime and users. Experimenters can configure it as desired. UltraFlow Access resources have also been provided with collectd-write-oml2 so that, instead of being stored locally, measurements can be reported to an OML database, UltraFlow local one for the moment, and Fed4Fire OML database in the future.

## 5. USING THE TESTBED

The Emulab web interface gives UltraFlow the option of creating local users and gives them access to configure experiments and reserve resources. In addition, UltraFlow Access also accepts users from Fed4Fire trusted identity providers who can configure experiments, reserve and access its resources via Fed4Fire tools, such as JFed [12]. A user registered in the federation can use Fed4Fire tools, reserve resources on multiple testbeds and interconnect them. In order to use this tool, users can get an account from iMinds authority provider and download their certificates from there since JFed requires this user certificate and its password to be able to log in. Once connected, users can directly start experimenting or perform any configuration change: add/remove collected measurement options, change OML2 endpoint or select local measurement storage, install new software, change optical card parameters, etc.

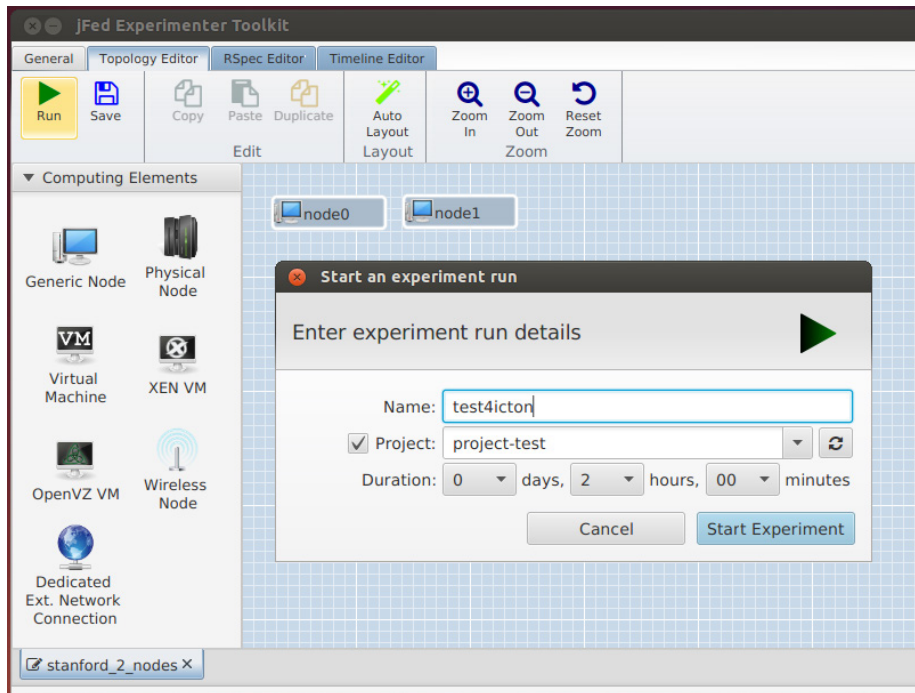


Figure 3. Usage of JFed to reserve two nodes in UltraFlow Access testbed at Stanford University.

## 6. CONCLUSIONS

This paper described an approach for the integration of an optical networking testbed in a federation of testbeds using state of the art software, following the Fed4Fire approach. The limitations of virtualization and programmability of testbed devices can easily be overcome by co-locating additional computers and installing software on them that implement most APIs developed in GENI and FIRE. Emulab is a useful tool for this purpose. Resource management, experiment control, measurement collection and testbed monitoring get installed but the overall operation of opening a testbed requires quite a few resources.

The challenge of most virtualised environments is getting beyond functional validation of the concepts being tested on them. However, the ability to map virtual machines on physical machines can enable valid performance measurements and hence it is important to adopt frameworks that support this capability.

## ACKNOWLEDGEMENTS

The research of this paper was partially financed by the European Union's FP7 grant agreement no. 318389 Fed4FIRE Project, the National Science Foundation (grant no. 111174), NSERC, the Spanish projects CRAMnet (grant no. TEC2012-38362-C03- 01) and TIGRE5-CM (grant no. S2013/ICE-2919). The authors would also like to acknowledge the support of the Chair of Excellence of Bank of Santander at UC3M.

## REFERENCES

- [1] V.W.S. Chan, "Optical flow switching networks," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1079-1091, 2012.
- [2] Z. Rosberg, J. Li, F. Li, and M. Zukerman, "Flow scheduling in optical flow switched (OFS) networks under transient conditions," *Journal of Lightwave Technology*, vol. 29, no. 21, pp. 3250-3264, 2011.
- [3] L.G. Kazovsky, A.R. Dhaini, M. De Leenheer, T.S. Shen, S. Yin, and B.A. Detwiler, "UltraFlow access networks: A dual-mode solution for the access bottleneck," in *Proc. International Conference on Transparent Optical Networks (ICTON'13)*, Cartagena, Spain, Jun. 2013.
- [4] D. Larrabeiti, J. A. Hernandez, I. Seoane, and R. Romeral, "Managing delay in the access," in *Proc. 17th European Conference on Networks and Optical Communications (NOC)*, pp. 1-8, 2012.
- [5] K. Obraczka, "Multicast transport protocols: A survey and taxonomy," *IEEE Communications Magazine*, vol. 36, no. 1, pp. 94-102, Jan. 1998.
- [6] Fed4FIRE (Federation for Future Internet Research and Experimentation) Research Project.  
<http://www.fed4fire.eu>
- [7] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "OMF: A control and management framework for networking testbeds," *ACM SIGOPS Operating Systems Review* 43 (4), 54-59, Jan. 2010.
- [8] [http://omf.mytestbed.net/projects/omf/wiki/OMF\\_Main\\_Page](http://omf.mytestbed.net/projects/omf/wiki/OMF_Main_Page)
- [9] <https://oml.mytestbed.net/projects/oml/wiki/>
- [10] <https://collectd.org>
- [11] <http://www.emulab.net>
- [12] <http://jfed.iminds.be>
- [13] O. Mehani, G. Jourjon, T. Rakotoarivelo, and M. Ott, "An instrumentation framework for the critical task of measurement collection in the future Internet," *Computer Networks*, vol. 63, pp. 68-83, Apr. 2014.